ICLR 2023 (notable top 5%)

IMAGE AS SET OF POINTS

Xu Ma, Yuqian Zhou, Huan Wang, Can Qin, Bin Sun, Chang Liu, Yun Fu

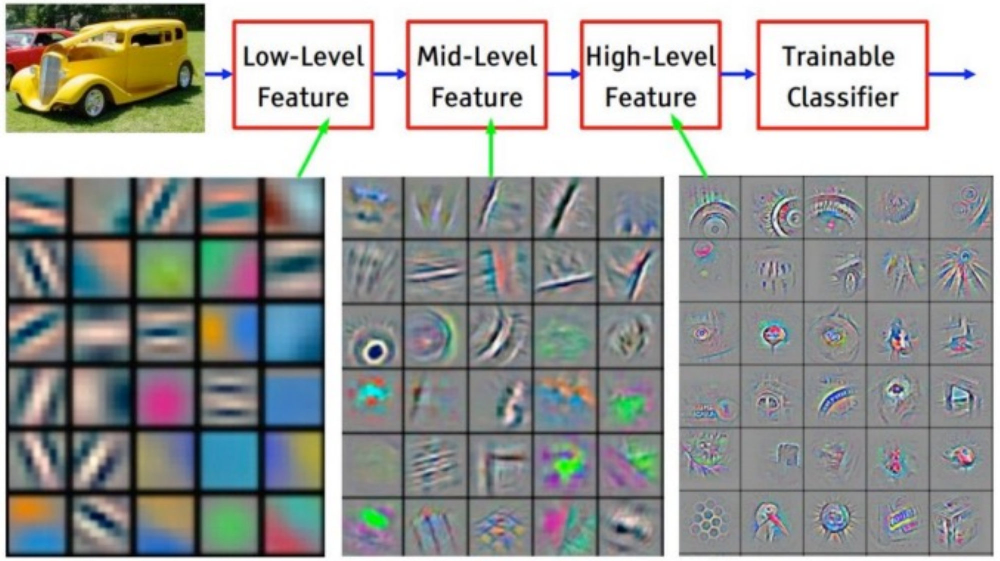
Efficient Learning Lab@POSTECH

Junwon Seo

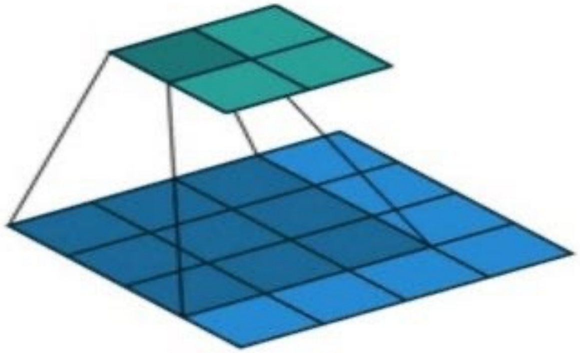
“The way we extract features depends a lot on how we interpret an image.”

Prevailing Methods

- Convolutional Neural Networks (ConvNets)



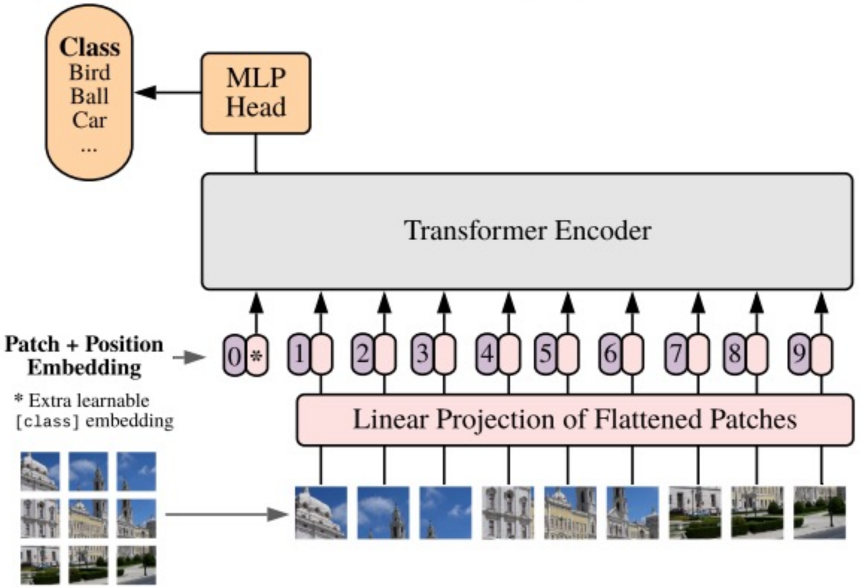
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]



- Image as **a collection of arranged pixels** in a rectangle form
- Benefiting from **locality** and **translation equivariance**

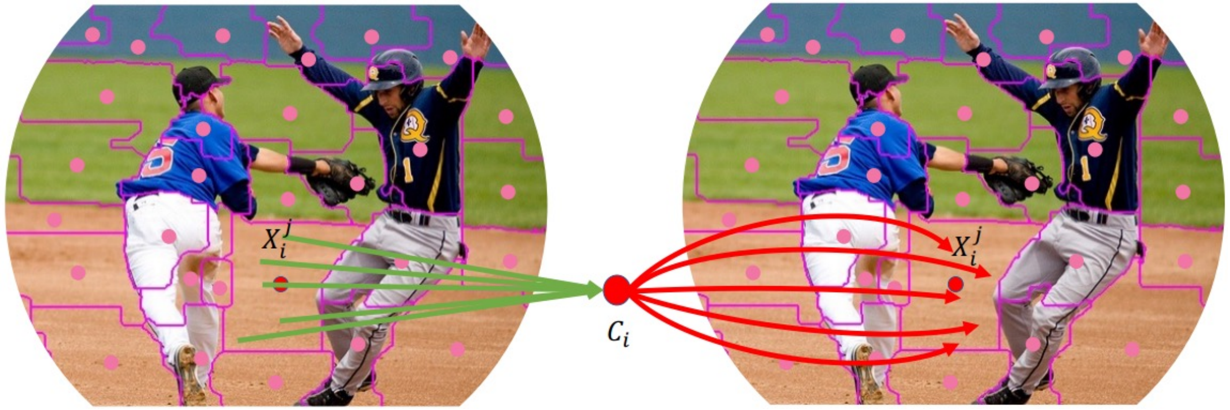
Prevailing Methods

- Vision Transformer(ViT)



- Image as **a sequence of patches**
- Self-attention operation

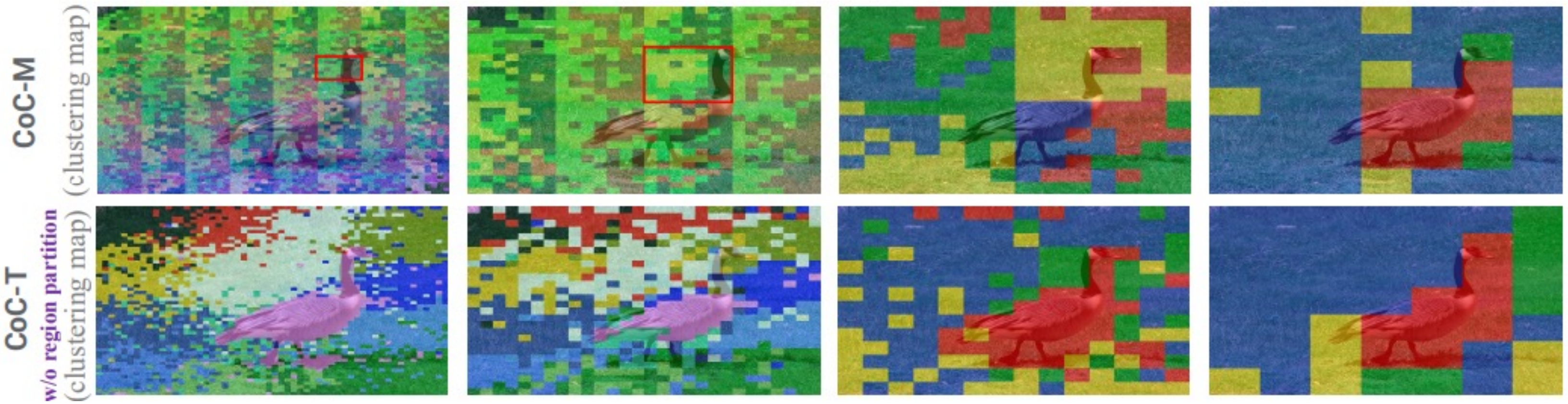
Context Cluster(CoC)



- Overview
  1. View image as a set of data points.
  2. Group all points into clusters
  3. Aggregate the points into a center
  4. Dispatch the center point to all points adaptively

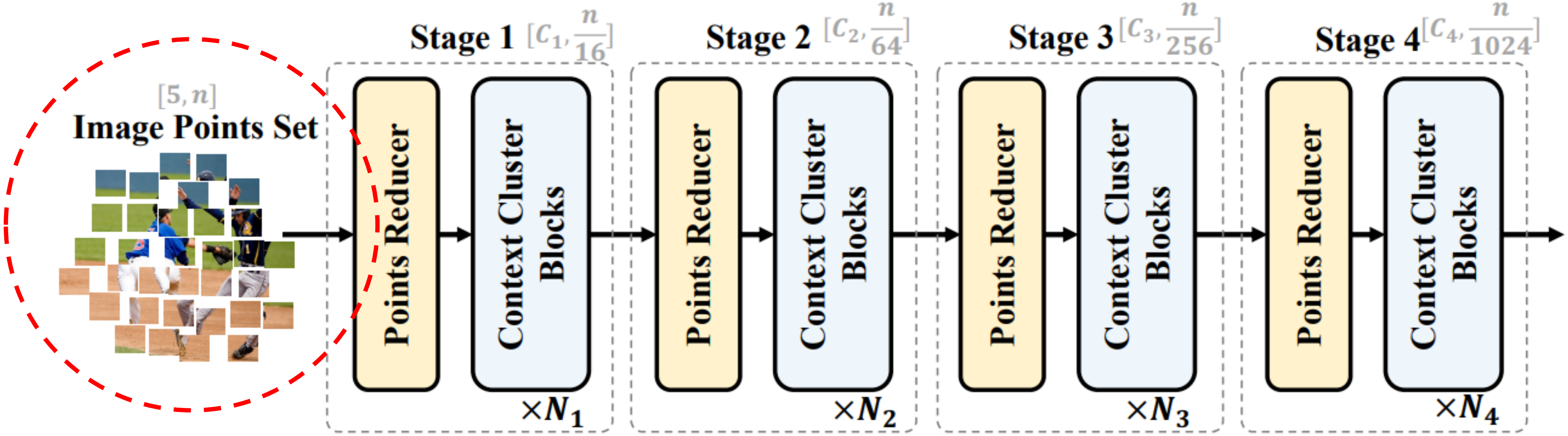
Expectations

- Generalization ability
  - In different domain, such as point clouds, RGBD images



- Interpretability
  - By visualizing the clustering in each layer, explicitly understand the learning

Context Cluster(CoC)



Given an input image $\boldsymbol{I} \in \mathbb{R}^{3 \times w \times h}$

2D coordinates of each Pixel: $\boldsymbol{I}_{i,j}$

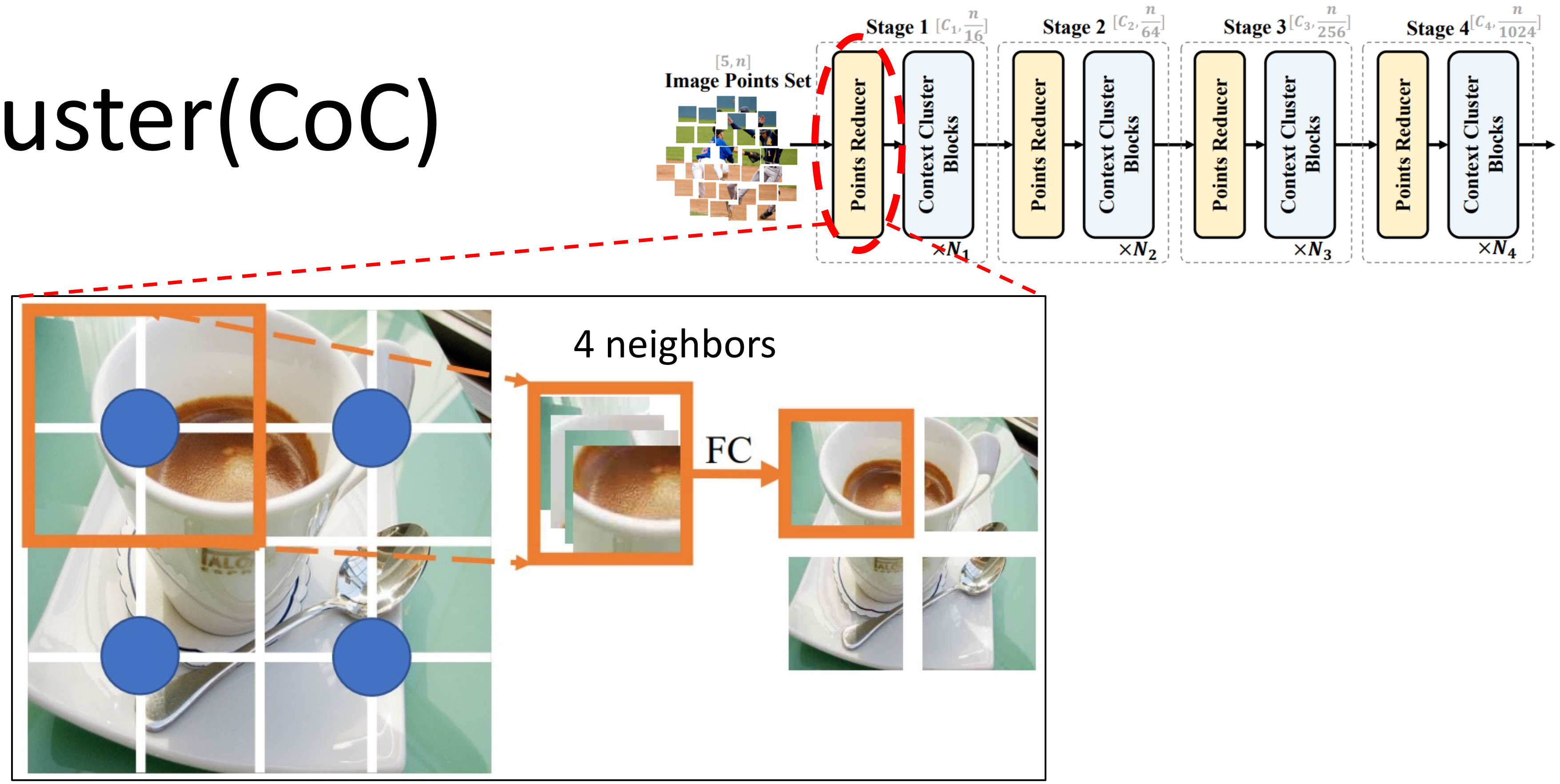
Coordinate is presented as $[\frac{i}{w} - 0.5, \frac{j}{h} - 0.5]$

Collection of points $\boldsymbol{P} \in \mathbb{R}^{5 \times n}$

(where $n = w \times h$)

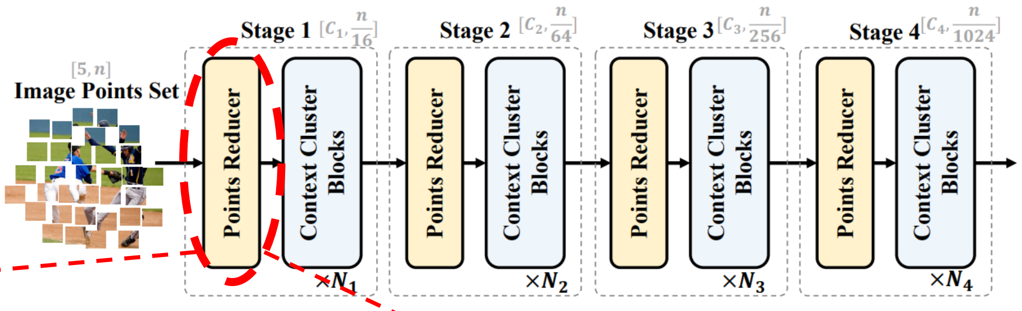
Each point contains color (r,g,b) and position (i, j)

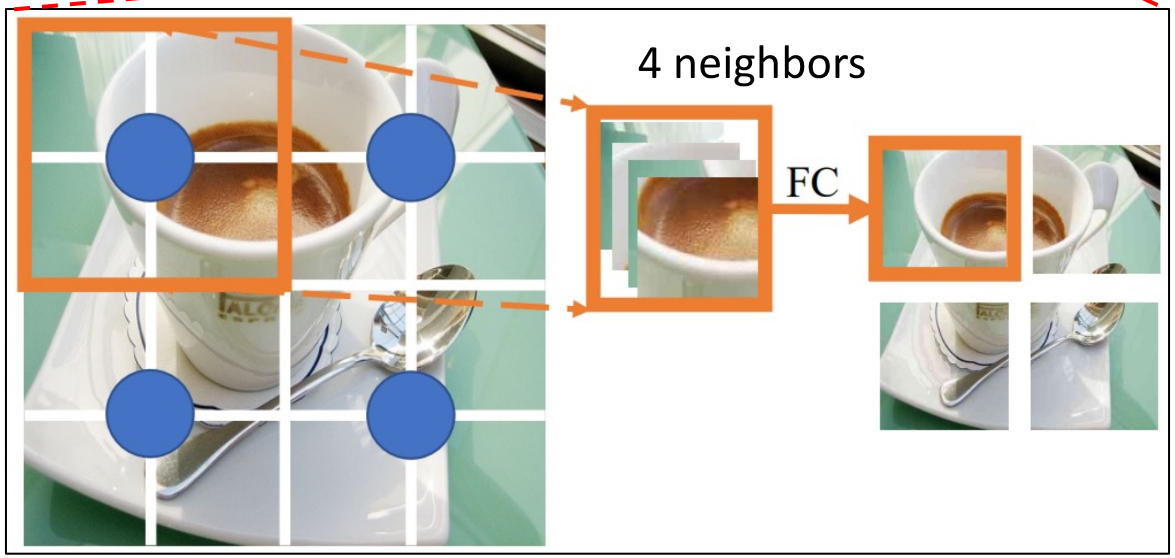
Context Cluster(CoC)



- All neighbors are concatenated along the channel dimension.
- FC layer is used to lower the dimensional number and fuse the information.

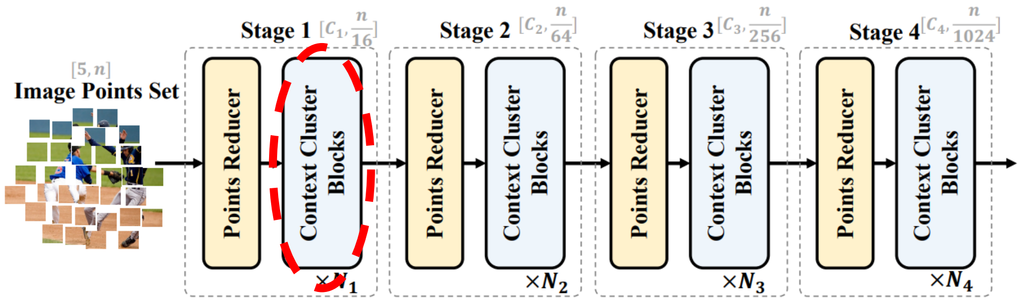
Context Cluster(CoC)

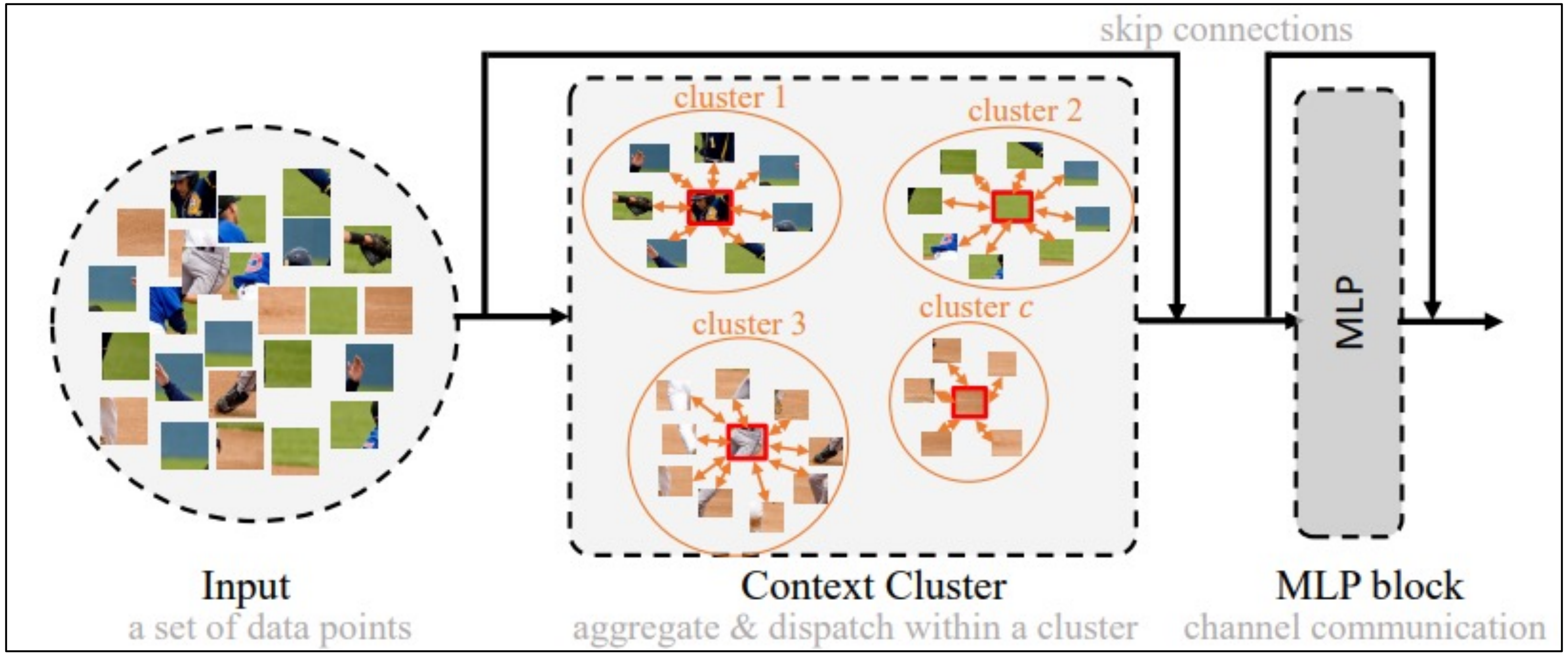




- Following the design of ConvNets and pyramid ViTs.
- The pooling operation is used in implementation.
- Avoid heavy indices search work

Context Cluster Block

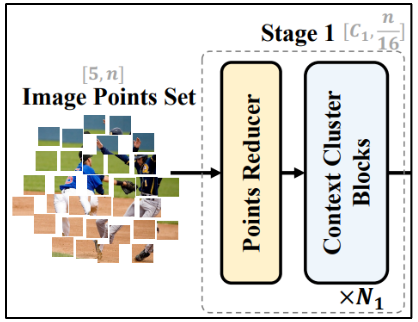


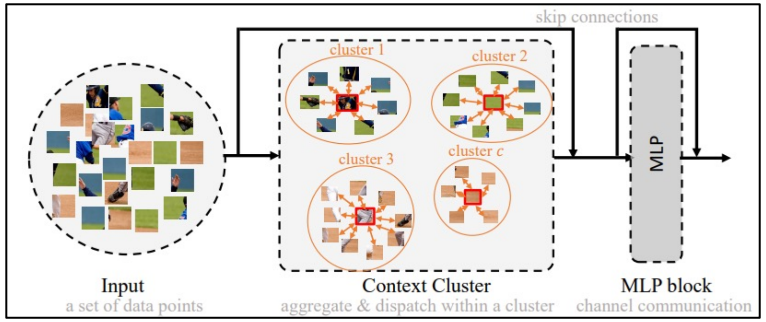


Context Cluster Box

Context Cluster Block

-Context Clustering-





$\boldsymbol{P} \in \mathbb{R}^{5\times n} \longrightarrow \boldsymbol{P} \in \mathbb{R}^{d\times n}$

1. Linearly Project $\boldsymbol{P} \rightarrow \boldsymbol{P_s}$ for similarity computation

   $(\boldsymbol{P} \in \mathbb{R}^{d\times n}, \boldsymbol{P_s} \in \mathbb{R}^{d'\times n})$

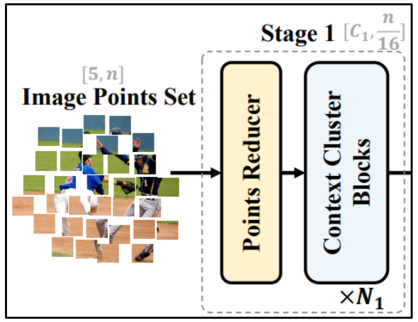
2. Evenly propose c centers in space
   - Center feature is computed by averaging its k nearest points[1]

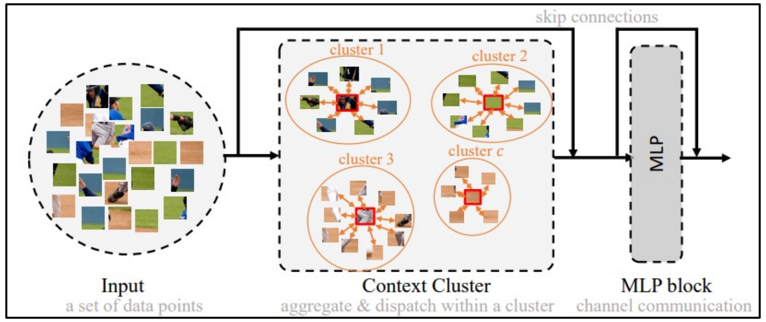
3. Calculate the pair-wise cosine similarity matrix $\boldsymbol{S} \in \mathbb{R}^{c\times n}$
   - Between $\boldsymbol{P_s}$ and the resulting set of center points
   - Each point contains both feature and position information
     - Implicitly highlight the points' distances(locality) and feature similarity
   - Some clusters may have zero points in extreme cases

1. Achanta et al. "SLIC superpixels compared to state-of-the-art superpixel methods"

Context Cluster Block

-Feature Aggregating-





$\boldsymbol{P} \in \mathbb{R}^{5\times n} \longrightarrow \boldsymbol{P} \in \mathbb{R}^{d\times n}$

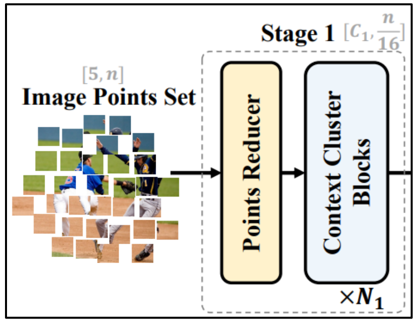
- Assuming a cluster contains m points (a subset in $\boldsymbol{P}$)
  - Similarity points and the cluster $s \in \mathbb{R}^m$ (a subset in $\boldsymbol{S}$)
  - Map the points to a value space to get $\boldsymbol{P}_v \in \mathbb{R}^{m\times d'}$ (Linearly Projected)
  - Projected Center v_c, Projected point v_i
- Aggregated feature $g \in \mathbb{R}^{d'}$ is given by

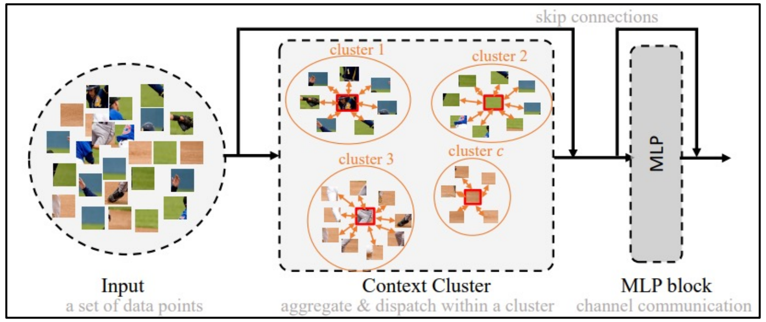
$$g = \frac{1}{C}\left(v_c + \sum_{i=1}^{m} \mathrm{sig}\left(\alpha s_i + \beta\right) * v_i\right), \qquad \text{s.t.,} \quad C = 1 + \sum_{i=1}^{m} \mathrm{sig}\left(\alpha s_i + \beta\right).$$

  - α and β are learnable scalars to scale and shift similarity
  - Sigmoid rescale the similarity to (0,1) (achieve much better results)

Context Cluster Block

-Feature Aggregating-





$\boldsymbol{P} \in \mathbb{R}^{5\times n} \longrightarrow \boldsymbol{P} \in \mathbb{R}^{d\times n}$

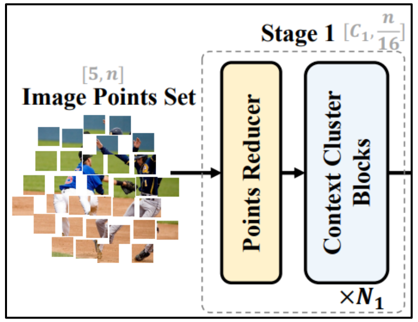
- Assuming a cluster contains m points (a subset in $\boldsymbol{P}$)
  - Similarity points and the cluster $s \in \mathbb{R}^m$ (a subset in $\boldsymbol{S}$)
  - Map the points to a value space to get $\boldsymbol{P}_v \in \mathbb{R}^{m\times d'}$ (Linearly Projected)
  - Projected Center v_c, Projected point v_i
- Aggregated feature $g \in \mathbb{R}^{d'}$ is given by

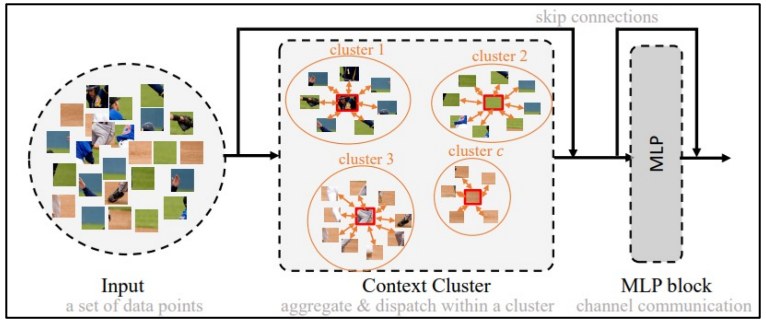
$$g = \frac{1}{C}\left(v_c + \sum_{i=1}^{m} \mathrm{sig}\left(\alpha s_i + \beta\right) * v_i\right), \qquad \text{s.t.,} \quad C = 1 + \sum_{i=1}^{m} \mathrm{sig}\left(\alpha s_i + \beta\right).$$

  - v_c emphasize the locality
  - 1 is added for numerical stability (if zero, not optimized ($1e^{-5}$ also doesn't help))

Context Cluster Block

-Feature Dispatching-





$\boldsymbol{P} \in \mathbb{R}^{5\times n} \longrightarrow \boldsymbol{P} \in \mathbb{R}^{d\times n}$

- Aggregated feature g is adaptively dispatched to each point in a cluster

$$p_i' = p_i + \mathrm{FC}\left(\mathrm{sig}\left(\alpha s_i + \beta\right) * g\right)$$

Skip connection

- Fully-connected(FC) Layer is for matching the feature dimension
  - d' -> d (original dimension)

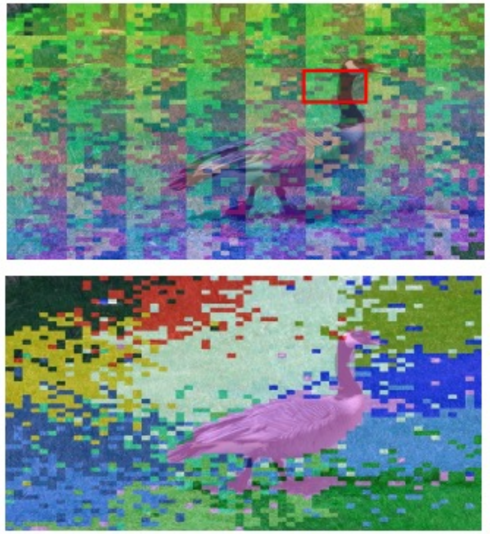
- By dispatching points, the points can communicate with one another and shares features from all points in the cluster

Results - Classification (ImageNet – 1K)

	Method	Param.	GFLOPs	Top-1	Throughputs (images/s)
MLP	♣ ResMLP-12 (Touvron et al., 2022)	15.0	3.0	76.6	511.4
	♣ ResMLP-24 (Touvron et al., 2022)	30.0	6.0	79.4	509.7
	♣ ResMLP-36 (Touvron et al., 2022)	45.0	8.9	79.7	452.9
	♣ MLP-Mixer-B/16 (Tolstikhin et al., 2021)	59.0	12.7	76.4	400.8
	♣ MLP-Mixer-L/16 (Tolstikhin et al., 2021)	207.0	44.8	71.8	125.2
	♣ gMLP-Ti (Liu et al., 2021a)	6.0	1.4	72.3	511.6
	♣ gMLP-S (Liu et al., 2021a)	20.0	4.5	79.6	509.4
Attention	♦ ViT-B/16 (Dosovitskiy et al., 2020)	86.0	55.5	77.9	292.0
	♦ ViT-L/16 (Dosovitskiy et al., 2020)	307	190.7	76.5	92.8
	♦ PVT-Tiny (Wang et al., 2021)	13.2	1.9	75.1	-
	♦ PVT-Small (Wang et al., 2021)	24.5	3.8	79.8	-
	♦ T2T-ViT-7 (Yuan et al., 2021a)	4.3	1.1	71.7	-
	♦ DeiT-Tiny/16 (Touvron et al., 2021)	5.7	1.3	72.2	523.8
	♦ DeiT-Small/16 (Touvron et al., 2021)	22.1	4.6	79.8	521.3
	♦ Swin-T (Liu et al., 2021b)	29	4.5	81.3	-
Convolution	♠ ResNet18 (He et al., 2016)	12	1.8	69.8	584.9
	♠ ResNet50 (He et al., 2016)	26	4.1	79.8	524.8
	♠ ConvMixer-512/16 (Trockman et al., 2022)	5.4	-	73.8	-
	♠ ConvMixer-1024/12 (Trockman et al., 2022)	14.6	-	77.8	-
	♠ ConvMixer-768/32 (Trockman et al., 2022)	21.1	-	80.16	142.9
Cluster	♥ Context-Cluster-Ti (ours)	5.3	1.0	71.8	518.4
	♥ Context-Cluster-Ti‡ (ours)	5.3	1.0	71.7	510.8
	♥ Context-Cluster-Small (ours)	14.0	2.6	77.5	513.0
	♥ Context-Cluster-Medium (ours)	27.9	5.5	81.0	325.2

- Comparable Performance
  - Even Better than baseline using a similar number of parameters and FLOPs.
- Obviously outperforms MLP
  - Not credited to MLP blocks
  - Contribute to the visual representation
- Cannot achieve SOTA
  - But proving the viability of a new feature extraction paradigm

Results - Classification (ImageNet – 1K)



Cluster	♥ Context-Cluster-Ti (ours)	5.3	1.0	71.8	518.4
	♥ Context-Cluster-Ti‡ (ours)	5.3	1.0	71.7	510.8
	♥ Context-Cluster-Small (ours)	14.0	2.6	77.5	513.0
	♥ Context-Cluster-Medium (ours)	27.9	5.5	81.0	325.2

- ‡ denotes a different region partition
- Performance differences are negligible
  - Demonstrate the robustness of CoC to the local region

Results – Visualization

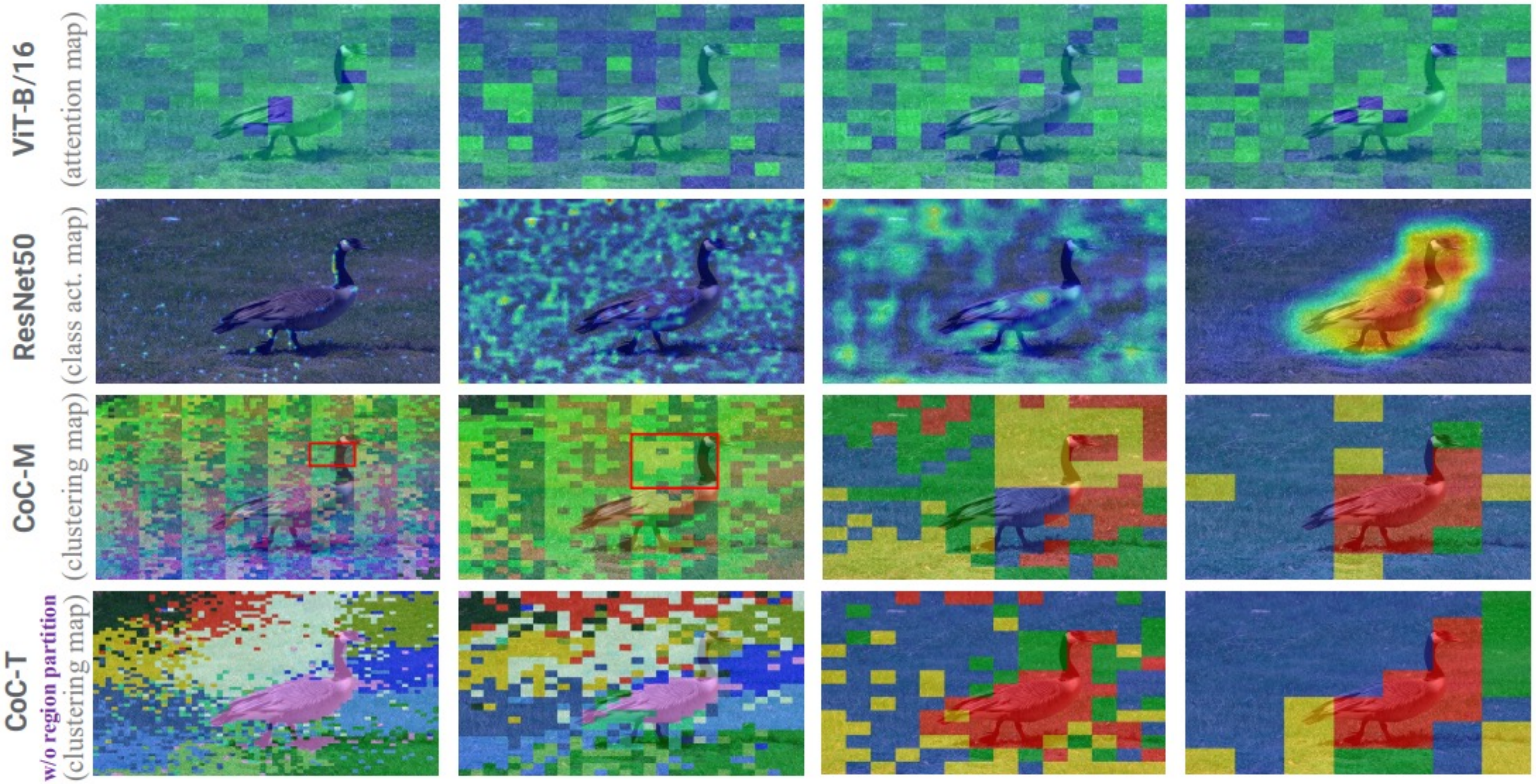


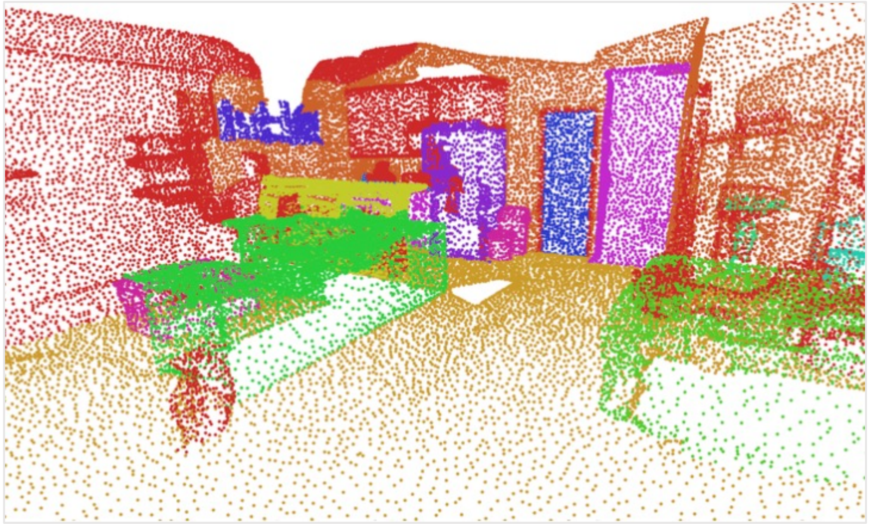
Figure 4: Visualization of activation map, class activation map, and clustering map for ViT-B/16, ResNet50, our CoC-M, and CoC-T without region partition, respectively. We plot the results of the last block in the four stages from left to right. For ViT-B/16, we select the [3rd, 6th, 9th, 12th] blocks, and show the cosine attention map for the `cls-token`. The clustering maps show that our Context Cluster is able to cluster similar contexts together, and tell what model learned visually.

- In the last stage, cluster goose as one object and background grass
- Can cluster similar context in very early stage.
- Most cluster emphasize the locality

Results – Point Cloud Classification

Table 3: Classification results on ScanObjectNN. All results are reported on the most challenging variant (PB_T50_RS).

Method	mAcc(%)	OA(%)
♠ SpiderCNN (Xu et al., 2018)	69.8	73.7
♠ DGCNN (Wang et al., 2019)	73.6	78.1
♠ PointCNN (Li et al., 2018)	75.1	78.5
♠ GBNet (Qiu et al., 2021)	77.8	80.5
♦ PointBert (Yu et al., 2022d)	-	83.1
♦ Point-MAE (Pang et al., 2022)	-	85.2
♦ Point-TnT (Berg et al., 2022)	81.0	83.5
♣ PointNet (Qi et al., 2017a)	63.4	68.2
♣ PointNet++ (Qi et al., 2017b)	75.4	77.9
♣ BGA-PN++ (Uy et al., 2019)	77.5	80.2
♣ PointMLP (Ma et al., 2022)	83.9	85.4
♣ PointMLP-elite (Ma et al., 2022)	81.8	83.8
♥ PointMLP-CoC (ours)	**84.4** $\uparrow 0.5$	**86.2** $\uparrow 0.8$



- Introduce PointMLP[1] as a foundation for our model
- Generalizability is most important.

Ma et al. “Rethinking network design and local geometry in point cloud”

Results – Object detection and segmentation

Table 4: COCO object detection and instance segmentation results using Mask-RCNN (1×).

Family	Backbone	Params	AP^{box}	AP^{box}_{50}	AP^{box}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
Conv.	♠ ResNet-18	31.2M	34.0	54.0	36.7	31.2	51.0	32.7
Attention	♦ PVT-Tiny	32.9M	36.7	59.2	39.3	35.1	56.7	37.3
Cluster	♥ CoC-Small/4	33.6M	35.9	58.3	38.3	33.8	55.3	35.8
	♥ CoC-Small/25	33.6M	**37.5**	**60.1**	**40.0**	**35.4**	**57.1**	**37.9**
	♥ CoC-Small/49	33.6M	37.2	59.8	39.7	34.9	56.7	37.0

- Table 4 shows that promising generalizability to downstream tasks.

Conclusion

- Introduction of a novel feature extraction paradigm for visual representation

- Image as a set of unorganized points and employ simplified clustering algorithms to extract features

- Achieves comparable or even better results than ConvNets and ViT baselines on multiple tasks and domains